

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
Факультет кібербезпеки, комп'ютерної та програмної інженерії  
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач випускової кафедри

\_\_\_\_\_ А.С. Савченко

« \_\_\_\_ » \_\_\_\_\_ 2021 р.

## ДИПЛОМНИЙ ПРОЕКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СПУПЕНЯ «БАКАЛАВР»

**Тема:** «Додаток тестування знань»

**Виконавець:** студент УС-412 Вареник Артем Миколайович  
(студент, група, прізвище, ім'я, по батькові)

**Керівник:** к. т. н., доцент Колісник Олена Василівна  
(науковий ступень, вчене звання, прізвище, ім'я, по батькові)

**Нормоконтролер:** ст. викл. Шевченко О.П.  
(П.І.Б.) (підпис)

КИЇВ 2021

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

Освітній ступінь: Бакалавр

Галузь знань, спеціальність, спеціалізація: 12 “Інформаційні технології”,  
122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

А.С. Савченко

«\_\_\_»\_\_\_\_\_ 2021 р.

## ЗАВДАННЯ

**на виконання дипломного проекту студента**

Вареник Артем Миколайович

(прізвище, ім'я, по батькові)

1. Тема проекту: «Додаток тестування знань» затверджена наказом ректора № 636/ст. від 22.04.2021р.
2. Термін виконання роботи: з 10.05.2021 по 20.06.2021р.
3. Вихідні дані до роботи: розробка додатку тестування знань за допомогою мови програмування Java.
4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): аналіз існуючих програм тестування, порівняльний аналіз мов програмування і середовищ розробки, розробка додатку тестування знань, відображення розробленого додатку тестування знань, висновки.
5. Перелік обов'язкового графічного матеріалу: слайди MS PowerPoint.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Етапи виконання дипломного проекту	Термін виконання	Примітка
1	Аналіз літератури та джерел за темою дипломного проекту.	10.05.21р. – 12.05.21р.	
2	Розробка та затвердження плану дипломного проекту.	13.05.21р.	
3	Проведення консультації з науковим керівником щодо створення першого розділу.	14.05.21р.	
4	Огляд і порівняльний аналіз існуючих програм тестування знань.	15.05.21р. – 18.05.21р.	
5	Порівняльний аналіз мов програмування та середовищ розробки.	19.05.21р. – 22.05.21р.	
6	Налаштування середовища розробки.	23.05.21р.	
7	Розробка інтерфейсу користувача.	24.05.21р. – 28.05.21р.	
8	Реалізація модуля тестування.	29.05.21р. – 03.06.21р.	
9	Розробка допоміжного функціоналу.	04.06.21р. – 06.06.21р.	
10	Завершення оформлення пояснювальної записки.	07.06.21р. – 09.06.21р.	
11	Підготовка презентації до захисту.	10.06.21р. – 12.06.21р.	
12	Підготовка до захисту дипломного проекту.	13.06.21р. – 14.06.21р.	

Студент

( *Вареник А.М.* )

Керівник дипломної роботи

( *Колісник О.В.* )

## **РЕФЕРАТ**

Пояснювальна записка до дипломного проекту «Додаток тестування знань» містить: 53 сторінок, 15 рисунків, 2 таблиці, 13 літературних джерел.

Об'єкт дослідження: автоматизація процесу тестового контролю і перевірки знань.

Предмет дослідження: програмні продукти для проведення тестового контролю і перевірки знань.

Мета роботи: розробити програмний продукт, що дозволяє проведення тестового контролю знань за допомогою технічних засобів.

Методи дослідження, технічні та програмні засоби: порівняльний аналіз, обробка літературних джерел.

Отримані результати та їх новизна: створено програму, яка автоматизує процес тестового контролю і перевірки знань. Розроблений додаток може слугувати для розширення можливостей контролю і оцінювання рівня знань, а також для підвищення якості освітніх процесів в навчальних закладах України.

ТЕСТУВАННЯ, КОНТРОЛЬ ЗНАНЬ, МОВИ ПРОГРАМУВАННЯ, СЕРЕДОВИЩА РОЗРОБКИ, ІНТЕРФЕЙС КОРИСТУВАЧА.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ .....	7
ВСТУП.....	8
РОЗДІЛ 1. ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	11
1.1. Аналіз існуючих додатків .....	11
1.1.1. Програма MyTestX.....	11
1.1.2. Програма Айрен.....	13
1.1.3. Програма EasyQuizzy .....	14
1.2. Вибір мови програмування .....	15
1.2.1. Мова програмування Java.....	15
1.2.2. Мова програмування C# .....	18
1.2.3. Мова програмування Python.....	19
1.2.4. Мова програмування C++.....	21
1.2.5. Порівняльний аналіз мов програмування .....	23
1.3. Вибір середовища розробки.....	25
1.3.1. Eclipse .....	25
1.3.2. IntelliJ IDEA .....	26
1.3.3. Порівняльний аналіз середовищ розробки .....	26
Висновки до розділу 1 .....	28
РОЗДІЛ 2. РОЗРОБКА ДОДАТКУ ТЕСТУВАННЯ ЗНАНЬ.....	29
2.1. Налаштування і підготовка середовища розробки .....	29
2.2. Опис функціональності системи .....	30
2.3. Опис основних етапів розробки .....	32
2.3.1. Розробка інтерфейсу користувача .....	32
2.3.2. Розробка модуля тестування .....	38
2.3.3. Розробка допоміжного функціоналу .....	39
Висновки до розділу 2 .....	42

РОЗДІЛ 3. ВІДОБРАЖЕННЯ РОЗРОБЛЕНОГО ДОДАТКУ ТЕСТУВАННЯ	
ЗНАНЬ .....	44
3.1. Інструкція з використання програмного продукту .....	44
Висновки до розділу 3 .....	49
ВИСНОВКИ.....	50
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	52

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Фреймворк (англ. Framework)	– програмне забезпечення, що полегшує розробку і об'єднання різних компонентів великого програмного проекту
ООП	– об'єктно-орієнтоване програмування
ОС	– операційна система
ПК	– персональний комп'ютер
API (Application Programming Interface)	– набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення
GC (Garbage Collection)	– одна із форм автоматичного керування пам'яттю
GUI (Graphical User Interface)	– графічний інтерфейс користувача
IDE (Integrated Development Environment)	– інтегроване середовище розробки
JAR (Java ARchive)	– архів в якому міститься частина Java програми
JDK (Java Development Kit)	– комплект розробника застосунків Java
JRE (Java Runtime Environment)	– мінімальна реалізація віртуальної машини
JVM (Java Virtual Machine)	– віртуальна машина для виконання Java програм
PEP (Python Enhancement Proposal)	– стандарт написання коду на Python
XML (Extensible Markup Language)	– мова для опису даних за допомогою тегів

## ВСТУП

Перевірка знань є однією з важливих складових частин в навчальному процесі. Існує велика кількість методів перевірки і форм контролю рівня знань, найпоширенішими з них є: письмовий, усний і графічний контроль, практична перевірка та методи самооцінки та самоконтролю. Ці методи хоча і є ефективними, але, зазвичай, займають дуже багато часу і не завжди дозволяють оцінити всіх.

Тому важливим питанням постало, як максимально заощадити час і оцінити всіх студентів протягом відведеного часу, для цього було винайдено нові методи: тестовий або програмований контроль.

Останнім часом ці методи набули широкого застосування в освітніх закладах України, адже їх використання є ефективним для контролю і перевірки знань. Однією з причин, набору популярності даних методів зумовлено входженням України у Болонський процес до визначальних рис якого відноситься тестова модульно-рейтингова система контролю знань.

Тестовий контроль являє собою ефективний метод перевірки і контролю рівня знань за допомогою системи спеціально підготовлених тестів, які повинні відповідати певним вимогам. Такий вид контролю дуже часто використовується для актуалізації певних знань, визначення рівня засвоєння студентом певної теми, а також інколи на заліках та іспитах.

В загальному розумінні, тест – це вибір однієї правильної відповіді з множини відповідей, що запропоновані до питання.

Виділяють такі переваги тестового контролю:

- можна охопити майже всі етапи процесу навчання;
- однакові умови для всіх хто проходить тестування;
- простота і швидкість процедури проведення;
- економія часу і можливість тестування великої кількості студентів одночасно;
- підвищення об'єктивності оцінювання знань;



- можливість використання в процесі проведення тестування ПК;
- швидкість перевірки і оцінювання студента.

Але також тестовий контроль має і недоліки:

- метод дає можливість випадково вибрати правильну відповідь на питання методом виключення;
- не можливо дослідити логіки міркувань студента;
- витрачається багато часу і зусиль для розробки тестових завдань.

Застосування інформаційних технологій значно підвищує ефективність процесу навчання і робить можливим автоматизувати певні його складові частини.

Тестування можливо проводити як в паперовій так і в електронній формі. Як було сказано вище, однією з головних переваг є те, що в процесі тестового контролю можливо використовувати різноманітні технічні засоби.

Комп'ютерне тестування – тестування, яке автоматизоване за допомогою спеціалізованих програмних продуктів. Воно розширює можливості контролю та оцінювання рівня знань і є альтернативою традиційним методам перевірки і форм контролю, що згадувались вище. Даний метод має ряд переваг у порівнянні із традиційним:

- метод дозволяє знизити витрати на проведення тестування;
- надає можливість застосовувати в процесі тестування мультимедійні можливості ПК;
- обробка результатів проводиться автоматично, що збільшує об'єктивність оцінювання.

**Актуальність теми** полягає у необхідності створення відповідних програмних засобів або додатків для проведення тестового контролю знань. Ці програми слугують для підвищення ефективності проведення тестування і дозволяють звести до нуля час перевірки цих тестів, адже надають результат відразу після закінчення тестування.

**Мета дипломного проекту** – розробити програмний продукт, що дозволяє проведення тестового контролю знань за допомогою технічних засобів.

Для досягнення мети у дипломному проекті визначено такі задачі:

- проаналізувати існуючі додатки тестування;
- огляд мов програмування і середовищ розробки та здійснити їх порівняльний аналіз;
- огляд фреймворків для реалізації інтерфейсу користувача;
- реалізувати інтерфейс користувача;
- розробити модуль тестування.

**Об’єктом дослідження** є автоматизація процесу тестового контролю і перевірки знань.

**Предметом дослідження** є програмні продукти для проведення тестового контролю і перевірки знань.

**Практичне значення** дипломного проекту полягає у створенні додатку тестування знань, який спростить процес проведення тестового контролю і підвищить його ефективність.

# РОЗДІЛ 1

## ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 1.1. Аналіз існуючих додатків

Майже всі популярні програмні продукти, які забезпечують процес тестування складаються з декількох функціональних модулів, що можуть використовуватися, як окремі незалежні програми так і об'єднуватися в єдине ціле і представляти повноцінну систему. До складу таких систем, зазвичай, входять такі модулі:

- модуль тестування, що і забезпечує процес проходження тесту;
- модуль для створення і модифікації тестів;
- модуль, що забезпечує обробку і аналіз результатів тестування;
- модуль, що забезпечує тестування мережею.

Розглянемо найбільш популярні системи комп'ютерного тестування, які часто використовуються в процесі навчання.

#### 1.1.1. Програма MyTestX

MyTestX – це програма для створення і проведення тестування, яка складається з декількох функціональних модулів, які можуть складати цілу систему або працювати як незалежний виконуваний файл. Вона складається з модуля для тестування, редактора та журналу результатів [1].

Перша версія програми була розроблена ще у 2003 році й являла собою простий і зручний додаток для тестування з невеликими функціональними можливостями. Поступово MyTestX доповнювався новими можливостями та функціоналом і на сьогодні це не просто додаток, а цілий комплекс програм, який використовується для підготовки і проведення тестування.

Кафедра КІТ (47)				НАУ 21 28 76 000 ПЗ			
Виконав	Вареник А.М.			Вибір програмного забезпечення	Літера	Аркуш	Аркушів
Керівник	Колісник О.В.					11	18
Консульт.					412 122		
Н-котрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						

Найактуальнішою версією на даний момент є MyTestXPro 11.0, в якій можна працювати з 10-ю типами завдань: від вибору однієї правильної відповіді, до перестановки букв і, навіть, заповнення пропусків.

В одному тесті, відповідно може містити завдання як одного типу так і є можливість використовувати завдання різних типів.

Кожне питання може містити до десяти варіантів відповідей. В тексті питань і відповідей можна використовувати як звичайний текст, який можна по-різному відформатувати, так і малюнки та формули.

Однією з корисних функцій, які надає MyTestX – це можливість створення паперового варіанту тесту. Це дуже корисно, коли немає можливості пройти тест на комп'ютері.

Програма доступна у 2-х версіях: безкоштовна (MyTestX) з обмеженим функціоналом і розширена (MyTestXPro), яка містить в собі весь доступний функціонал. Але за розширену версію потрібно буде заплатити після тестового періоду.

До переваг MyTestX можна віднести:

- простий і зрозумілий інтерфейс;
- можливість створення паперового тесту;
- підтримка великої кількості типів завдань;
- можна організувати як локальне тестування так і мережею;
- можна створювати різні види тестів;
- має різні режими тестування.

А до недоліків:

- працює тільки на ОС Windows;
- за розширену версію потрібно заплатити;
- відсутній імпорт даних з інших форматів (txt, rtf, xml та інші);
- відсутність автоматичного збереження результату без участі викладача.

### 1.1.2. Програма Айрен

Айрен – це програма для створення і проведення тестування для перевірки та контролю знань. Назва пішла від скорочення Interactive Remote Education Network (IREN) [2].

Проведення тестування можливо, як на одному комп'ютері, так і локальною мережею і навіть через Інтернет. Програма складається з двох модулів, які можна використовувати окремо один від одного: мережевої версії і програми редагування.

Мережева версія слугує для створення тестів і їх проведення через мережу й збереження і обробку результатів. А за допомогою редактора тестів створюють тести, які потім можна зберегти й використовувати, як окремий автономний виконуваний файл, який можна запустити на будь-якому комп'ютері з ОС Windows для проходження тестування, але в даному випадку результат збережено не буде.

Айрен дозволяє створювати тести з різними типами завдань. Можна створити як завдання з вибором однієї або декількох правильних відповідей, введення відповіді за допомогою клавіатури так і на встановлення певної відповідності.

До переваг Айрен можна віднести:

- можливість створення різноманітних типів завдань;
- підтримка імпорту питань з текстового файлу;
- можливість тестування локальною мережею та через Інтернет;
- безкоштовна для завантаження та використання;
- створення тестів у вигляді окремих файлів;
- можна вести моніторинг процесу проведення тестування.

Програма також має і недоліки:

- результат зберігати доводиться вручну, а не автоматично;
- працює тільки на ОС Windows;
- складність налаштування;
- результати зберігаються в архіві.

### 1.1.3. Програма EasyQuizzzy

EasyQuizzzy – програма для створення і редагування тестів. Кожен створений тест являє собою незалежну програму, яку можна запустити на будь-якому комп'ютері з ОС Windows і пройти тестування [3].

Текст питань і відповідей можна по різному форматувати і вставляти в них картинки, формули, діаграми, різні символи та інше.

В програмі підтримуються різні типи завдань для тестування: вибір однієї або декількох правильних відповідей, вільна відповідь за допомогою клавіатури, альтернативний вибір, встановлення правильної послідовності та інші. Тест може складатися з завдань одного типу, а також і вміщати в себе комбінацію декількох або навіть всіх типів.

EasyQuizzzy дає можливість користувачу налаштувати такі параметри тесту, як кількість питань і їх порядок, критерії і систему оцінювання, а також можливо задати обмеження часу на проходження тесту і виведення правильних відповідей після кінця тестування.

Також можна створити підсумковий звіт проходження тесту, який потім можна зберегти і роздрукувати. У звіті можливо включити відображення питань з правильними і неправильними відповідями, питань без відповідей і в разі, якщо на питання дано неправильну відповідь, то вивести правильну.

Створюванні тести шифруються, що робить неможливим перегляд питань і відповідей іншими програмами.

Перевагами EasyQuizzzy є:

- кожен тест – це окрема незалежна програма;
- можливість обмеження часу проходження тесту;
- створення підсумкового звіту;
- шифрування тестів.

Недоліки можна виділити наступні:

- є платною;
- створенні тести сумісні тільки з ОС Windows.

## 1.2. Вибір мови програмування

Мова програмування – це мова для розробки різноманітних програм, де програмісти описують певні алгоритми і правила для виконання комп’ютером. На даний момент налічують близько 3 тисяч мов програмування, більшість з них, звісно, вже не використовуються, а інші, навпаки, активно розвиваються і доповнюються новими можливостями.

Тому через велику кількість мов, перед створенням програмного продукту постає питання, яку мову програмування краще вибрати для реалізації. Від цього вибору може залежати дуже багато: від простоти і швидкості створення програмного продукту до можливості перенесення його на інші платформи. Але, обов’язково, потрібно розуміти, що ідеальної мови програмування не існує і в кожній є свої переваги і недоліки, які звісно ж будуть впливати на процес розробки.

### 1.2.1. Мова програмування Java

Java – одна з найпопулярніших мов програмування, яку відносять до об’єктно-орієнтованих. Вона була розроблена американською компанією “Sun Microsystems”, яка спеціалізувалась на виробництві різного програмного і апаратного забезпечення. В 1991 році мова була названа “Oak” і використовувалась виключно для програмування побутових електронних приладів. А вже в 1995 році назву було змінено на “Java” на честь сорту кави, що ми й можемо побачити на логотипі, де розміщена чашка кави. Велику роль в розробці відіграв канадський інженер Джеймс Гослінг.

Головними цілями при розробці Java, було те, що потрібно створити мову з подібним синтаксисом як в C/C++, але в той же час він повинен бути більш “простим, об’єктно-орієнтованим та звичним” [4].

На сьогодні Java являє собою платформу і екосистему, яка об’єднує різноманітні технології, за допомогою яких можна створити від простого десктопного додатку до веб-сервісів і великих серверних додатків у великих корпораціях.

Також на Java створюють програмне забезпечення для різних пристроїв, таких як: персональний комп'ютер, смартфони і навіть побутова техніка. Наприклад, на ОС Android велика кількість програм написано саме на ній.

Однією з важливих особливостей Java є принцип WORA (write once, run anywhere), який був реалізований в мові. Суть полягає в тому, що код написаний на Java, буде працювати на будь-якій платформі з установленою JRE, в той час, коли на інших мовах програмування буде потрібно написати код під кожен платформу. Кросплатформність досягається за допомогою компіляції коду, що написаний на Java в байт-код, який потім виконується на віртуальній машині Java, яка не залежить від платформи і є частиною JRE. Більш детально і наглядно як це працює зображено на рис. 1.1.

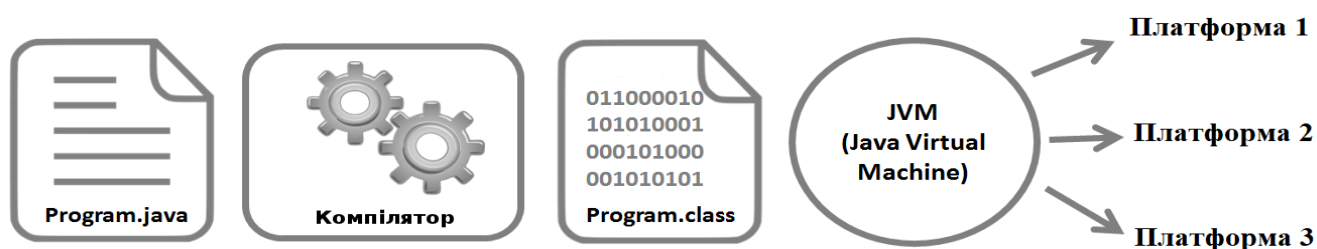


Рис. 1.1. Принцип WORA

Java забезпечує механізм обміну даними та програмами між декількома комп'ютерами, що дає нам приріст продуктивності та ефективності системи. Також за допомогою технології CORBA, ми можемо обмінюватися різними об'єктами в розподіленому середовищі.

Ще однією з особливостей Java є те, що в ній реалізований механізм керування пам'яттю, який називають автоматичний збирач сміття (Garbage Collector). Таким чином, програміст створює об'єкт, а JVM очищує пам'ять за допомогою GC, коли об'єкти перестають бути потрібним. Це може відбуватися в будь-який час, але найкраще, коли програма не використовується, адже цей процес може призводити до короткого зависання.



Java є надійною мовою програмування, що забезпечується двома принципами: ООП і сильна типізація.

Java об'єктно-орієнтована мова програмування, а отже вона підтримує основні принципи ООП: абстракція, поліморфізм, наслідування, інкапсуляція.

Всі дії і дані групуються в класи об'єктів. Всі ці об'єкти є похідними і успадковують властивості і базову поведінку від головного об'єкта. Створена ієрархія наслідування збільшує читабельність коду і зменшує кількість невимушених помилок при його написанні.

Сильна або строга типізація являє собою один із варіантів роботи з типами даних, що використовуються в мові програмування. Java відноситься до мов програмування з сильною типізацією, а отже програмісту доводиться виконувати більший обсяг роботи, щоб присвоїти кожній змінній, константі або виразу строго певний тип, але в той же час, дані інтерпретуються однозначно, що зменшує шанс некоректної роботи програми.

Можна виділити основні переваги Java:

- написане програмне забезпечення на одній платформі, може бути запущене на іншій;
- простота синтаксису, що прискорює процес вивчення мови;
- перевіряє код під час компіляції;
- дозволяє написати програму в якій можуть виконуватися декілька операцій паралельно;
- програми на мові Java дешево розробляти та обслуговувати;
- автоматичне керування пам'яттю;
- надійність.

Недоліками можна назвати:

- не дивлячись на різні способи оптимізації, Java є ресурсномісткою і повільною;
- не забезпечує можливості резервного копіювання;
- написані коди на Java великі за об'ємом, що зменшує читабельність.

### 1.2.2. Мова програмування C#

C# – це проста, сучасна, з безпечною системою типізації об'єктно-орієнтована мова програмування, яка була розроблена корпорацією Microsoft під керівництвом Андреса Гейлсберга і Скота Вільтамота в 1998 – 2001-х роках. Символ “#”, який міститься в назві можна трактувати, як дві пари плюсів, що в свою чергу натякає на новий етап у розвитку C# порівняно з C++.

Сьогодні мова програмування C# вважається одною з найпотужніших та швидко розвиваючих в IT-галузі. Завдяки зручному Cі-подібному синтаксису, великій кількості бібліотек і фреймворків, на ній пишуть найрізноманітніші програми: мобільні додатки, комп'ютерні програми, веб-сервіси, веб-сайти, ігри та багато чого іншого.

C# також називають “очищеною версією Java”, адже приблизно 70% синтаксичних можливостей він перейняв саме в неї. Ще 10% у C++ і 5% у Visual Basic, а решту 15% це і є унікальність мови – реалізація ідей розробників.

Оскільки C# об'єктно-орієнтована мова програмування, то вона підтримує велику кількість корисних функцій: інкапсуляцію, поліморфізм, наслідування, а також перевантаження операторів, статичну типізацію, вказівники на функції-члени класів та багато іншого. А так як C# все ще активно розвивається, то з кожною новою версією розробники додають багато цікавих функцій, наприклад, асинхронні методи, динамічне зв'язування [5].

В C# як і в Java реалізований збирач сміття, який керує в автоматичному режимі виділенням і звільненням пам'яті для додатку від об'єктів, які не використовуються.

Також в мові є інструмент обробки виключень, за допомогою якого програміст може виявляти і обробляти помилки в коді. Ця функція допомагає впоратися з непередбаченими проблемами, які можуть виникнути при використанні програми.

C# є мовою програмування з сильною типізацією в якій діє загальна система роботи з типами. Це означає, що кожна константа і змінна має тип. Мова не дозволяє звернення до неініційованих змінних, що виключає можливість виходу за межі певного масиву даних і виконання безконтрольного приведення типів при роботі програми.

Щоб програми на C# працювали необхідним і достатнім є те, щоб була встановлена і налаштована платформа .NET Framework, яка вбудована в пакет Windows, але і існують версії для Linux і MacOS. Написаний код після компіляції перетворюється в проміжну мову, яка в свою чергу під час виконання програми на комп'ютері перетворюється на машинний код. Це означає що мова C#, як і Java кросплатформна і її виконавчі файли можуть бути запуснені без перекомпіляції на тій чи іншій платформі.

Перевагами C# є:

- всі типи даних, які використовуються в мові мають фіксований розмір, що в свою чергу збільшує її “мобільність” і спрощує процес програмування;
- автоматичне виділення і очищення пам'яті від об'єктів, які не використовуються;
- синтаксис мови є простим і зрозумілим;
- активно розвивається і доповнюється функціонал;
- за допомогою платформи Xamarin, можуть бути написані програми для iOS, Android, MacOS і Linux;
- C# об'єднує в собі найкращі ідеї таких мов програмування, як Java, C++, Visual Basic;
- кросплатформність.

Недоліки:

- орієнтація в основному іде на Windows платформу;
- безкоштовна тільки для людей, які навчаються та індивідуальних програмістів, а для великої команди ліцензія обійдеться у велику суму;
- C# сильно залежить від .NET Framework.

### 1.2.3. Мова програмування Python

Python являє собою потужну об'єктно-орієнтовану мову програмування для загального призначення. Вона була розроблена нідерландським програмістом Гвідо ван Россумом і випущена в 1990 році. До 2018 року він активно брав участь у її розвитку та оптимізації.

Назва мови була запозичена в серіалі “Літаючий цирк Монті Пайтона”, переглядати який полюбляв Гвідо ван Россум.

На сьогоднішній час, Python легко конкурує з іншими мовами програмування, адже активно розвивається і з кожним роком стає все більш популярнішою мовою. Це обумовлено тим, що це універсальна мова і може використовуватися для вирішення великої кількості різноманітних задач.

За допомогою Python можуть бути написані скрипти по автоматизації, ігри, веб-ресурси, мобільні додатки та інше. Активно використовується при автоматизації тестування і для створення вбудованих систем для великої кількості різноманітних пристроїв, таких як банкомати, термінали. Також великої популярності має в галузі математики, адже може використовуватися для обробки великого об’єму даних та виконання складних математичних обрахунків.

Такі компанії як Spotify і Amazon використовують Python для аналізу даних і створення рекомендацій, Disney для створення анімацій, а NASA для проведення наукових обрахунків.

Однією з переваг Python є те, що він має простий і логічний синтаксис, в якому було прибрано все зайве. Це робить його дуже подібним до англійської мови, що збільшує читабельність і сприйняття коду. Цей синтаксис дозволяє програмісту написати програму з меншою кількістю вихідного коду ніж на інших мовах програмування.

Python – мова програмування з строгою динамічною типізацією, тобто тип змінної буде визначатися в момент надання значення, але при зміні значення може змінюватися і тип даних, що спрощує написання коду новачку і в подальшому дає змогу уникнути безліч помилок і багів.

Ще однією перевагою Python є те, що існує велика кількість фреймворків і бібліотек, що спрощують вирішення складних задач.

Також перевагою Python є те, що вона інтерпретована, тобто написаний код буде виконуватися рядок за рядком і якщо буде знайдена помилка – виконання зупиниться і буде повідомлено про неї. Python показує лише одну помилку, тобто першу, яка була виявлена, навіть якщо програма має декілька помилок. Це набагато

економить час при розробці і полегшує подальше налагодження програми. Інтерпретатор мови існує для багатьох популярних платформ, що дозволяє написаний код на одній платформі запустити цей же код на іншій [6].

Оскільки Python підтримує модульність, то написаний код може бути розділений на модулі, які потім можна використовувати заново вже в інших програмах. Також існує велика кількість колекцій стандартних модулів, які надають різну функціональність: від введення-виведення даних до створення графічного інтерфейсу користувача.

Перевагами Python можна виділити:

- легкий для читання та вивчення;
- Python має бібліотеки які дозволяють інтегрувати з іншими мовами програмування, як C/C++ та Java;
- широке застосування в різних галузях;
- стандарт для написання коду PEP, що забезпечує читабельність коду при переході від одного розробника до іншого;
- активно розвивається і розширюється новими колекціями бібліотек і фреймворків;
- кросплатформна мова;
- Open Source, що дозволяє будь-кому взяти участь в розвитку і покращенню мови.

Недоліки:

- через те, що код виконується рядок за рядком це призводить до зменшення продуктивності;
- погана реалізація багатопоточності;
- велике споживання пам'яті і ресурсів;
- залежний від системних бібліотек.

#### 1.2.4. Мова програмування C++

C++ – потужна мова програмування загального призначення, яка підтримує декілька парадигм: узагальнене, об'єктно-орієнтоване та процедурне програмування.

Більше уваги було приділено підтримці першим двом. Мова була розроблена датським програмістом Б'ярном Страуструпом в 1980-х роках. І вважалась доповненням до мови C для розширення її можливостями об'єктно-орієнтованого програмування. Через це мову почали називати "C with classes". З часом популярність мови ставала все більшою і в неї почали добавляти нові можливості, які робили її не просто доповнення, а вже повністю самостійною мовою програмування з назвою "C++" [7].

У 1990-х роках C++ стає однією з найпопулярніших мов програмування, для розробки різноманітного програмного забезпечення, операційних систем, відеоігор, драйверів, клієнтських і серверних програм та іншого. Наприклад, ОС Windows в більшій частині написана саме на цій мові.

Останнім часом C++ набирає популярність в написанні мобільних додатків, а також і у веб-розробці – для створення веб-додатків або сервісів, що їх обслуговують. Тому C++ має широке використання і дає можливість написати майже будь-які види програм.

Одним із головних аргументів чому C++ став і є популярним на даний час, це його швидкість. Він займає лідируючі місця в рейтингах по цьому параметру, адже компіляція, як і сам час виконання програми на C++ набагато швидша, ніж у більшості інших мов програмування загального призначення. Тому, якщо затримка є критичною метрикою, а продуктивність дуже важливою, то C++ є найкращим вибором для розробки. Такі вимоги постійно виникають на серверах із великим навантаженням і обробкою великої кількості даних, таких як веб-сервери, сервери баз даних та інші.

Однією з особливостей можна назвати те, що C++ надає повний контроль над керуванням пам'яттю, але це може розглядатися як перевагою так і недоліком, адже це збільшує відповідальність користувача за управління пам'яттю, а не за тим, щоб нею керував збирач сміття.

В C++ написаний вихідний код перетворюється компілятором в виконуваний файл, який містить набір машинних інструкцій, а наявність під всі популярні платформи бібліотек та компіляторів дозволяє компілювати один вихідний код на

різних платформах. Але кожна платформа має свої особливості, що може принести деякі труднощі, однак якщо не використовувати, при написанні коду, специфічні властивості і функції певної операційної системи, то програми на C++ в більшій мірі володіють переносимістю і кросплатформністю.

C++ є мовою програмування, яка дуже близька до апаратного забезпечення, ніж більшість інших мов, що робить її дуже корисною, де програмне забезпечення тісно пов'язане з апаратним забезпеченням і на рівні програмного забезпечення потрібна підтримка низького рівня. Це забезпечує можливість працювати на низькому рівні з адресами, пам'яттю, портами тощо. Тому різні вбудовані системи, такі як медичні машини, розумні годинники, безпілотні автомобілі, сенсори тощо, використовують C++, як основну мову програмування.

Перевагами C++ є:

- швидкість роботи написаних на C++ програм;
- близькість до апаратного забезпечення і робота на низькому рівні;
- широке застосування;
- велика кількість бібліотек, які спрощують розробку;
- підтримка декількох парадигм програмування;
- повний контроль над керуванням пам'яті.

Недоліки:

- нема підтримки вбудованих потоків;
- погана підтримка модульності;
- відсутність збирача сміття;
- зазвичай використовується для написання під конкретну платформу.

#### 1.2.5. Порівняльний аналіз мов програмування

Для проведення порівняльного аналізу, виділимо базові вимоги до мови програмування – критерії, що будуть важливі для розробки додатку. Порівняємо вище оглянуті мови за такими критеріями:

- підтримка ООП;
- ручне керування пам'яттю;

- збір сміття;
- типізація;
- підтримка виключень;
- швидкість виконання програми;
- читабельність коду;
- простота: наскільки легко працювати і використовувати бібліотеки;
- графічний інтерфейс користувача (GUI): наскільки легко працювати з GUI;
- 2D графіка: простота використання 2D графіки і її швидкість;
- кросплатформність: легкість перенесення і запуску програми на різних платформах (Windows, Linux, MacOS та інші).

Результат порівняння представимо у вигляді таблиці 1.1.

Деякі критерії оцінимо за шкалою від 0 до 10, де 0 – дуже погано, 5 – добре, а 10 – відмінно.

Таблиця 1.1

#### Порівняльний аналіз мов програмування

Критерії\Мова	Java	C#	Python	C++
Підтримка ООП	+	+	+	+
Ручне керування пам'яттю	-	-	-	+
Збір сміття	+	+	+	-
Типізація	Статична сильна явна	Статична сильна явна	Динамічна сильна неявна	Статична слабка явна
Підтримка виключень	+	+	+	+
Швидкість виконання програми	6	7	3	8
Читабельність коду	7	7	8	6



Простота	6	5	9	4
Графічний інтерфейс користувача (GUI)	7	6	8	4
2D графіка	7	6	9	6
Кросплатформність	9	7	8	5

### 1.3. Вибір середовища розробки

Для розробки будь-якого програмного забезпечення, кожен програміст використовує інтегроване середовище розробки, що, зазвичай, включає в себе: редактор коду, компілятор або інтерпретатор і інструменти для відлагодження програми.

#### 1.3.1. Eclipse

Eclipse – модульне інтегроване середовище розробки, розроблене Eclipse Foundation. Дане середовище в основному написане на мові Java, що допомагає бути платформнонезалежним середовищем і працювати на різних платформах: Windows, Mac OS, Linux [8].

Переваги середовища:

- безкоштовне для використання;
- платформнонезалежне;
- підтримує велику кількість мов програмування для розробки;
- автоматичне збереження і відновлення сесії;
- підтримка різних функцій, утиліт і автодоповнення, що полегшує написання коду.

До недоліків можна віднести:

- використовує велику кількість системних ресурсів;
- не повна інтеграція з Maven;
- виникають труднощі з зворотною сумісністю плагінів;

- при встановленні можуть виникнути деякі труднощі;
- пошук необхідних функцій може зайняти багато часу.

### 1.3.2. IntelliJ IDEA

IntelliJ IDEA – інтегроване середовище розробки, розроблене компанією JetBrains для написання коду на різних мовах програмування. Існує два варіанта середовища: безкоштовне (Community Edition) і платного (Ultimate Edition), яка має більший функціонал. Перша версія була випущена в 2001 році і з того часу почала набувати популярності, як IDE з різноманітним набором інструментів, які підвищують ефективність розробки.

Перевагами середовища є:

- підтримка великої кількості мов програмування;
- навігація;
- простий та зручний інтерфейс;
- підтримка модуля візуального проектування;
- виявлення помилок при написанні коду;
- середовище пропонує вирішення тої чи іншої помилки;
- автодоповнення коду;
- відновлення сесії.

Недоліками є:

- використовує багато комп'ютерних ресурсів;
- виникають проблеми з продуктивністю при розробці великих проектів;
- повільне початкове завантаження середовища;
- за розширену версію потрібно буде заплатити.

### 1.3.3. Порівняльний аналіз середовищ розробки

Порівняємо вище оглянуті середовища розробки за такими критеріями:

- автодоповнення коду;
- плагіни;
- продуктивність;
- зручність використання;

- автоматичне створення коду;
- GUI designer;
- ціна.

Результат порівняння представимо у вигляді таблиці 1.2.

Таблиця 1.2

Порівняльний аналіз середовищ розробки

Критерії\Середовище	IntelliJ IDEA	Eclipse
Автодоповнення коду	Чудово реалізовано	Погано реалізовано
Плагіни	Підтримує близько 750	Підтримує близько 1250
Продуктивність	Працює набагато швидше і плавніше, коли мова йде про існуючий проект	Здатний обробляти великі та важкі проекти швидше завдяки індексації, яку він робить на початковій стадії проекту
Зручність використання	Простий та зручний інтерфейс	Інтерфейс зручний, але можуть виникнути труднощі при пошуку певних функцій
Автоматична генерація коду	Налаштовується і просте у використанні	Погано реалізовано
GUI designer	Вбудований	Вбудований
Ціна	Має безкоштовну версію (Community Edition) і платну (Ultimate Edition)	Безкоштовно

## Висновки до розділу 1

В першому розділі було розглянуто існуючі додатки для створення і проведення тестування, кожен з яких має свої переваги і недоліки. Провівши аналіз можна сказати, що найбільш зручною для використання є MyTestX.

Але основним недоліком розглянутих систем є те, що вони працюють тільки на ОС Windows і дещо функціонально перевантажені.

Також був зроблений аналіз мов програмування і визначено основні критерії, що будуть важливі для розробки додатку. При порівняльному аналізі, можна зробити висновок, що кожна мова в чомусь переважає, а в чомусь поступається іншим. Серед розглянутих мов програмування для реалізації додатку було вибрано Java, оскільки вона володіє потрібними і важливими особливостями, зокрема, додатки написані на Java не залежать від платформи і можуть бути використані на будь-якій платформі. Також в Java легше працювати з GUI і 2D графікою, що буде потрібно нам для розробки.

Відповідно вибраній мові програмування, було розглянуто два найпопулярніші середовища розробки. При виборі яких, можна зробити висновок, що середовищ розробки – безліч, серед яких є і безкоштовні і платні. Для розробки будемо використовувати IntelliJ IDEA, оскільки він потужний і в той же час простий у використанні і його функціонал повністю задовольняє потреби для розробки.

## РОЗДІЛ 2

### РОЗРОБКА ДОДАТКУ ТЕСТУВАННЯ ЗНАНЬ

#### 2.1. Налаштування і підготовка середовища розробки

Масштабні програмні продукти розробляють, зазвичай, за допомогою таких засобів, як Integrated Development Environment (IDE), які допомагають розробнику в написанні коду і прискорюють створення додатків. Тому розглянемо його налаштування і підготовку до використання.

Першим етапом для розробки додатку є налаштування і підготовка середовища в якому і буде здійснюватися написання і компіляція коду. Для реалізації додатку буде використовуватися IntelliJ IDEA, яка являється найпопулярнішою, на даний момент, середовищем розробки для Java.

Насамперед завантажимо IntelliJ IDEA з офіційного сайту. Середовище доступне у двох версіях – Ultimate (платна) і Community (безкоштовна). Для розробки функціоналу безкоштовної версії повністю достатньо [9].

Після процесу встановлення середовища – створюємо новий проект. Оскільки мовою програмування буде Java, для написання і запуску програм потрібно завантажити і встановити JDK – набір інструментів для розробки програмного забезпечення, що містить в собі компілятор (javac), Java Runtime Environment, стандартні бібліотеки мови, а також різну документацію [10].

Вигляд проекту в середовищі розробки можна побачити на рис. 2.1, де виділено основні елементи, якими користується розробник при написанні коду:

1. Структура проекту. Використовуючи цей елемент, можна переглянути деревоподібну структуру проекту. В папці .idea зберігаються всі налаштування проекту у вигляді набору xml-файлів. В папці src – папки і файли з вихідним кодом.

Кафедра КІТ (47)				НАУ 21 28 76 000 ПЗ			
Виконав	Вареник А.М.			Розробка додатку тестування знань	Літера	Аркуш	Аркушів
Керівник	Колісник О.В.					29	15
Консульт.							
Н-котрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						
					412 122		

2. Текстовий редактор з кодом програми. В цьому вікні можна переглядати і редагувати файли.
3. Вбудований в середу розробки командний рядок. За допомогою якого можна вводити аргументи і отримувати результати виконання програми.
4. Кнопка запуску компіляції проекту.

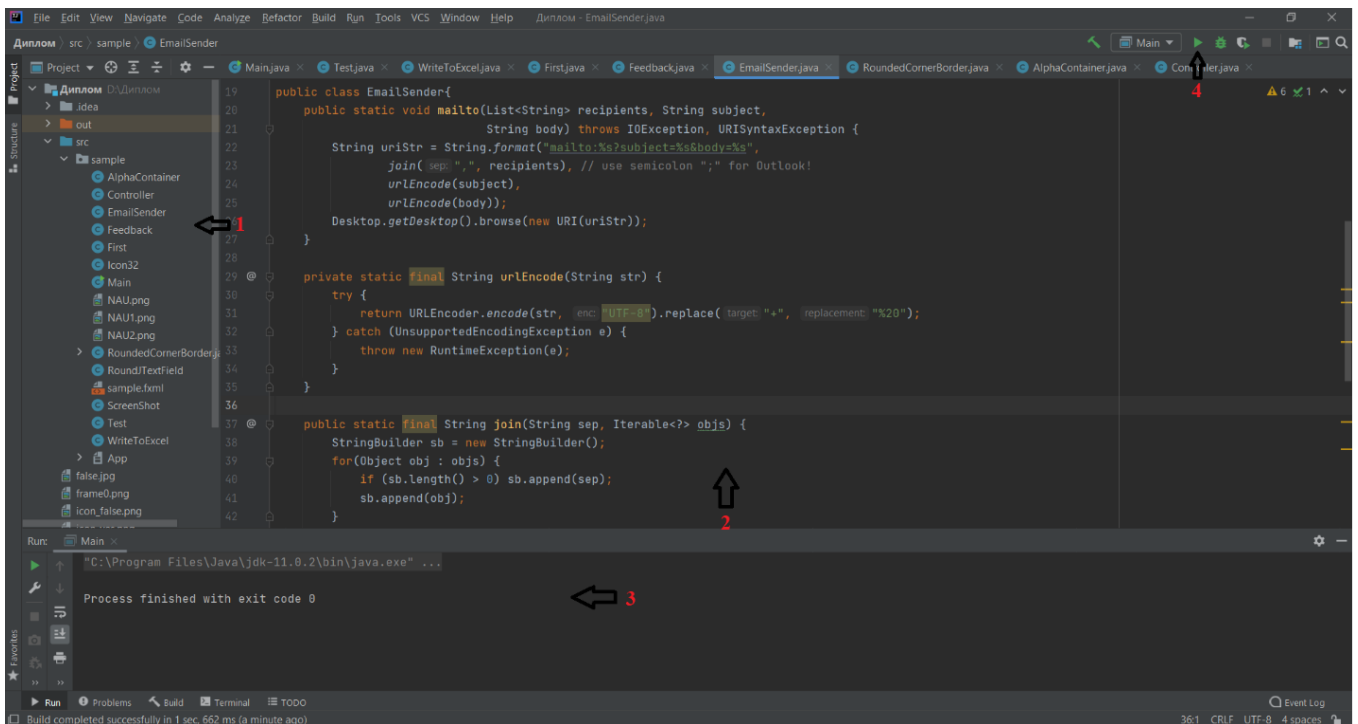


Рис. 2.1. Вікно проекту в IntelliJ IDEA

## 2.2. Опис функціональності системи

Опис бажаної функціональності системи можна представити за допомогою діаграми прецедентів, яка зображена на рис. 2.2. Діаграма являє собою множину акторів, прецедентів і асоціацій між ними та описує функціональні аспекти системи.

Актор – це будь-яка зовнішня сутність, наприклад, людина або будь-яка інша система, що працюють із системою і використовують її функціональні можливості для своїх потреб.

Прецедент – варіант використання – це виконувані системою дії або функції, які вона надає користувачеві – актору. Прецедент показує тільки “що” виконується, а не яким чином це буде виконано.

Система тестування знань включає в себе два актори: користувач і адміністратор, які взаємодіють із самою системою і один з одним.

Прецедентами є:

- введення даних користувача, що в свою чергу включає в себе ще два прецеденти: введення ПІБ і вибір факультету та групи;
- вибір білету;
- проходження тестування;
- надання зворотного зв'язку;
- технічна підтримка.

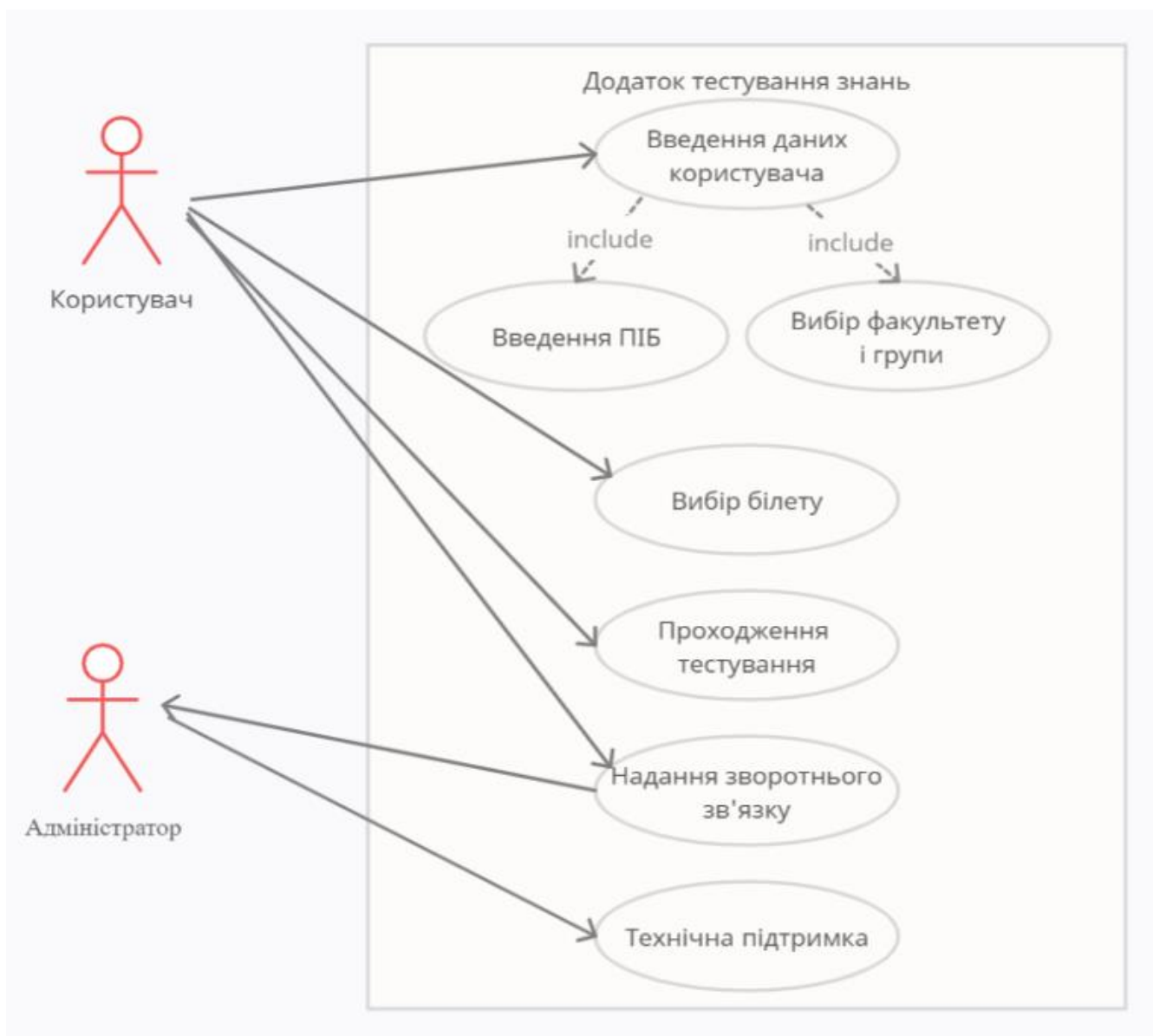


Рис. 2.2. Діаграма прецедентів системи

Користувач взаємодіє із системою за допомогою персонального комп'ютера, на якому встановлений додаток.

Адміністратор додатку надає необхідну технічну підтримку користувачеві, яка виникає при користуванні додатком. А також користувач і адміністратор можуть взаємодіяти за допомогою надання зворотного зв'язку.

### **2.3. Опис основних етапів розробки**

До основних етапів розробки додатку можна віднести:

- 1) Розробка інтерфейсу користувача. На даному етапі розробляється інтерфейс додатку, основних його вікон і компонентів, що реалізується за допомогою GUI фреймворків Java.
- 2) Розробка модуля тестування. Створюється алгоритм проходження тестування і на базі цього пишеться код.
- 3) Розробка допоміжного функціоналу. На даному етапі додаток доповнюється допоміжним функціоналом, який буде спрощувати користування ним.

#### **2.3.1. Розробка інтерфейсу користувача**

GUI (графічний інтерфейс користувача) складається з графічних компонентів управління (наприклад вікна, списки, кнопки та інше), за допомогою яких користувач може взаємодіяти з додатком.

Для реалізації графічного інтерфейсу користувача в Java використовують такі стандартні фреймворки як: бібліотеки Swing і AWT, а також, останнім часом, набуває популярності платформа JavaFX.

AWT (Abstract Windowing Toolkit) – це перший розроблений фреймворк для розробки графічного інтерфейсу користувача і віконних програм Java. Він визначає базовий набір елементів графічного інтерфейсу, зовнішній вигляд яких залежить від того, на якій платформі буде запущена програма, це означає, що вигляд всіх графічних компонентів визначається платформою, а не заздалегідь визначенні.

Це, зазвичай, призводить до того, що компоненти можуть виглядати і вести себе по-різному на різних платформах, що може викликати різні помилки. Тому



розробнику потрібно перевіряти коректність відображення графічних елементів на кожній платформі окремо.

Ієрархію класів Java AWT представлена на рис. 2.3, де ми можемо побачити, що існує дві групи елементів графічного інтерфейсу: компоненти і контейнери [11].

Компоненти – це елементарні сутності графічного інтерфейсу, а контейнери використовуються для зберігання цих компонентів.

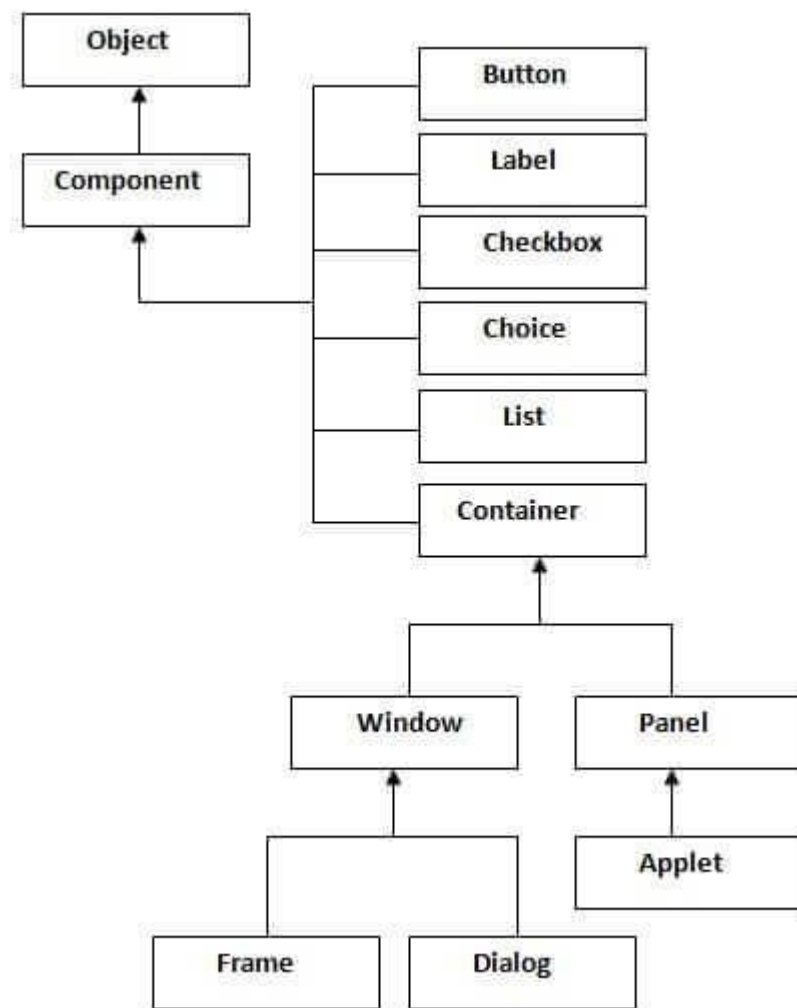


Рис. 2.3. Ієрархія класів Java AWT

AWT простий у використанні і має високу швидкодію, але має обмежений набір графічних компонентів і написана програма виглядає по-різному на різних платформах. Тому постало питання у знаходженні кращого підходу і на зміну AWT прийшов Swing.

Оскільки всі Swing-компоненти написані на Java, тому на відміну від AWT є платформнонезалежними і мають однакове відображення на різних платформах.

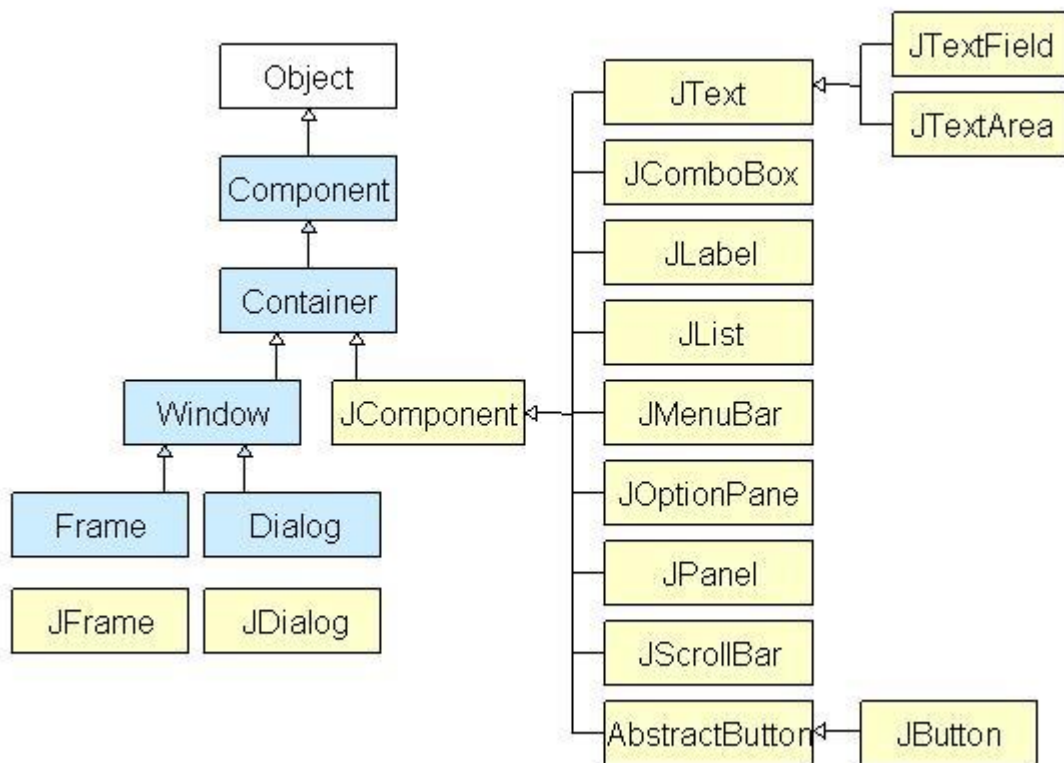


Рис. 2.4. Ієрархія класів Java Swing

Для відтворення графічних елементів використовується 2D графіка, що дозволяє збільшити різноманітність і функціональність компонентів, а також представлена можливість створювати нові компоненти, які можуть бути альтернативою основних компонентів.

Завдяки простоті використанні, гнучкості компонентів і великої кількості документації Swing став одним із популярніших графічних фреймворків Java.

Також однією з переваг Swing, яка спрощує створення графічного інтерфейсу – це вбудовані графічні редактори форм, майже, у всі середовища розробки.

Одним з таких графічних редакторів є IntelliJ GUI Designer, за допомогою якого можна створити графічний інтерфейс користувача на основі графічних компонентів бібліотеки Swing.

GUI Designer складається з таких основних компонентів (рис. 2.5):

- 1) Components Treeview – відображає ієрархічний вигляд компонентів, доданих у форму.
- 2) Property Editor – відображає властивості графічних компонентів, які можна редагувати.
- 3) Components Palette – містить усі доступні графічні компоненти, які можна вибрати для додавання у форму.
- 4) Form Workspace – графічний вигляд форми. Місце в яке перетягуються і розміщуються всі необхідні компоненти.

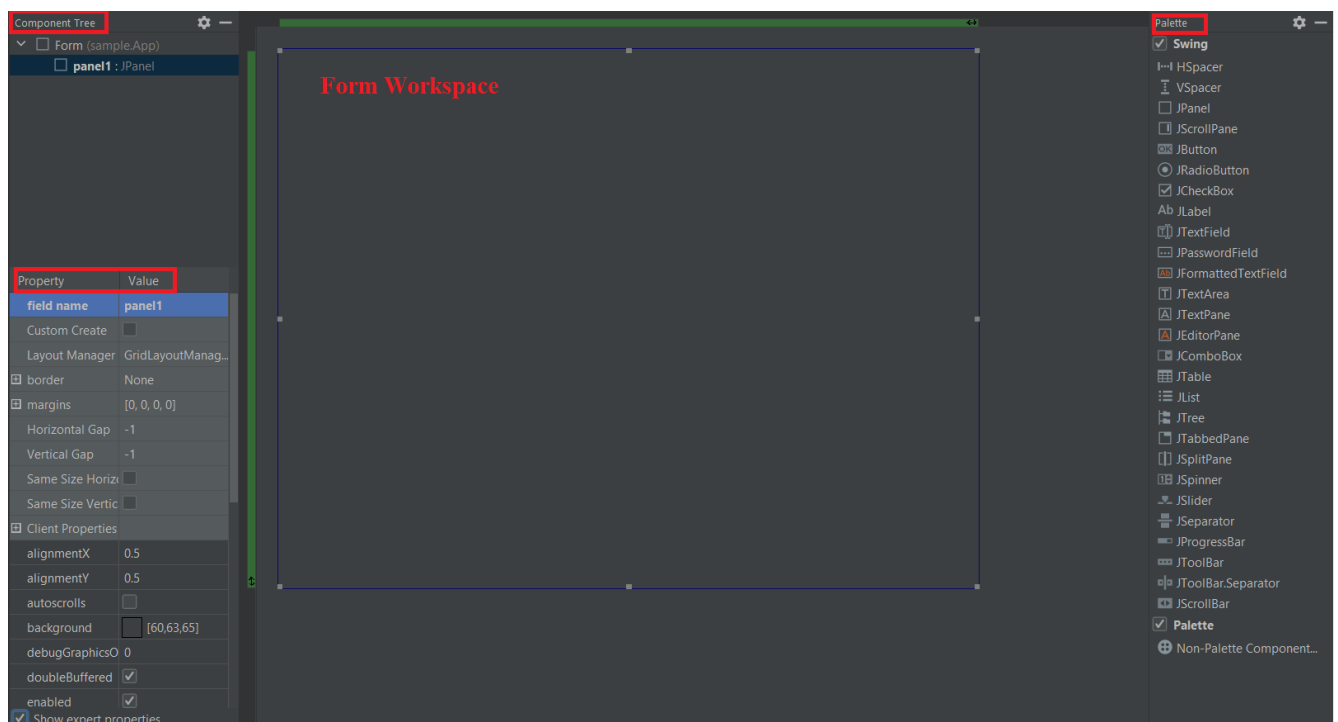


Рис. 2.5. GUI Designer

Працюючи з GUI Designer, робота відбувається в режимі дизайну, де перетягуються компоненти в форму, яка є контейнером для всіх інших графічних елементів необхідних для програми. Існує можливість, змінювати властивості цих елементів.

Відповідно до форми відбувається генерація коду в якому прописані всі компоненти і їх властивості, що ми використовуємо. Це пришвидшує розробку інтерфейсу, а саме: створення діалогових вікон та груп елементів керування, які використовуватимуться у контейнері верхнього рівня, такому як JFrame.

До головних вимог інтерфейсу користувача можна віднести наступне:

- послідовність;
- зрозумілість;
- логічність;
- забезпечення захисту від людських помилок;
- простота.

Для забезпечення цих вимог був розроблений інтерфейс додатку, який складається з декілька модулів – вікон:

- 1) Початкове вікно.
- 2) Вікно проходження тесту.
- 3) Вікно результату проходження тестування.
- 4) Вікно зворотного зв'язку.

Початкове вікно з'являтиметься, як тільки користувач запустить додаток і призначене для введення даних користувача, який буде проходить тестування. Вікно реалізується за допомогою таких Swing-компонентів: JFrame, JDialog, JLabel, JButton, JPasswordField, JCheckBox, JPanel, JComboBox.

Вікно має складатися з таких елементів:

- елементи для введення даних користувача;
- список білетів для тестування;
- перемикач для вибору наявності таймера в тестуванні;
- кнопка для початку тестування.

Наступний етап – створення вікна проходження тесту, яке слугує для тестування і буде наповнюватися даними відповідно до вибраного білету. Для реалізації використовуються такі Swing-компоненти: JFrame, JLabel, JTextArea, JTextField, JButton.

Вікно проходження тесту має складатися з таких елементів:

- номер поточного питання із загальної кількості в білеті;
- текстове поле з питанням;
- чотири варіанти відповіді;
- таймер (якщо він був вибраний перед початком тестування).

Після створення вікна проходження тестування, постає задача створення вікна відображення результату, де користувач зможе побачити результат і зробити за необхідністю знімок вікна. Вікно результату створюється за допомогою таких Swing-компонентів: JFrame, JDialog, JPanel, JLabel, JButton.

Для повноти відображення результату вікно має містити такі елементи:

- дані про користувача;
- кількість правильних і неправильних відповідей;
- оцінка за проходження тестування;
- кнопка для створення знімку вікна результату;
- кнопка для завершення тестування і закриття додатку.

Вікно зворотного зв'язку призначене для зв'язку з розробником шляхом відправлення повідомлення на пошту. Для реалізації використовувались такі Swing-компоненти: JFrame, JLabel, JButton, JTextArea, JComboBox.

Вікно має складатися з таких елементів:

- список тем звернень до розробника;
- текстове поле для введення повідомлення, яке буде відправлене розробнику;
- кнопка для відправлення повідомлення.

### 2.3.2. Розробка модуля тестування

Перед початком написання коду для модуля тестування, який і забезпечує сам процес проходження тесту користувачем, був розроблений алгоритм, який представлений за допомогою блок-схеми (рис. 2.6).

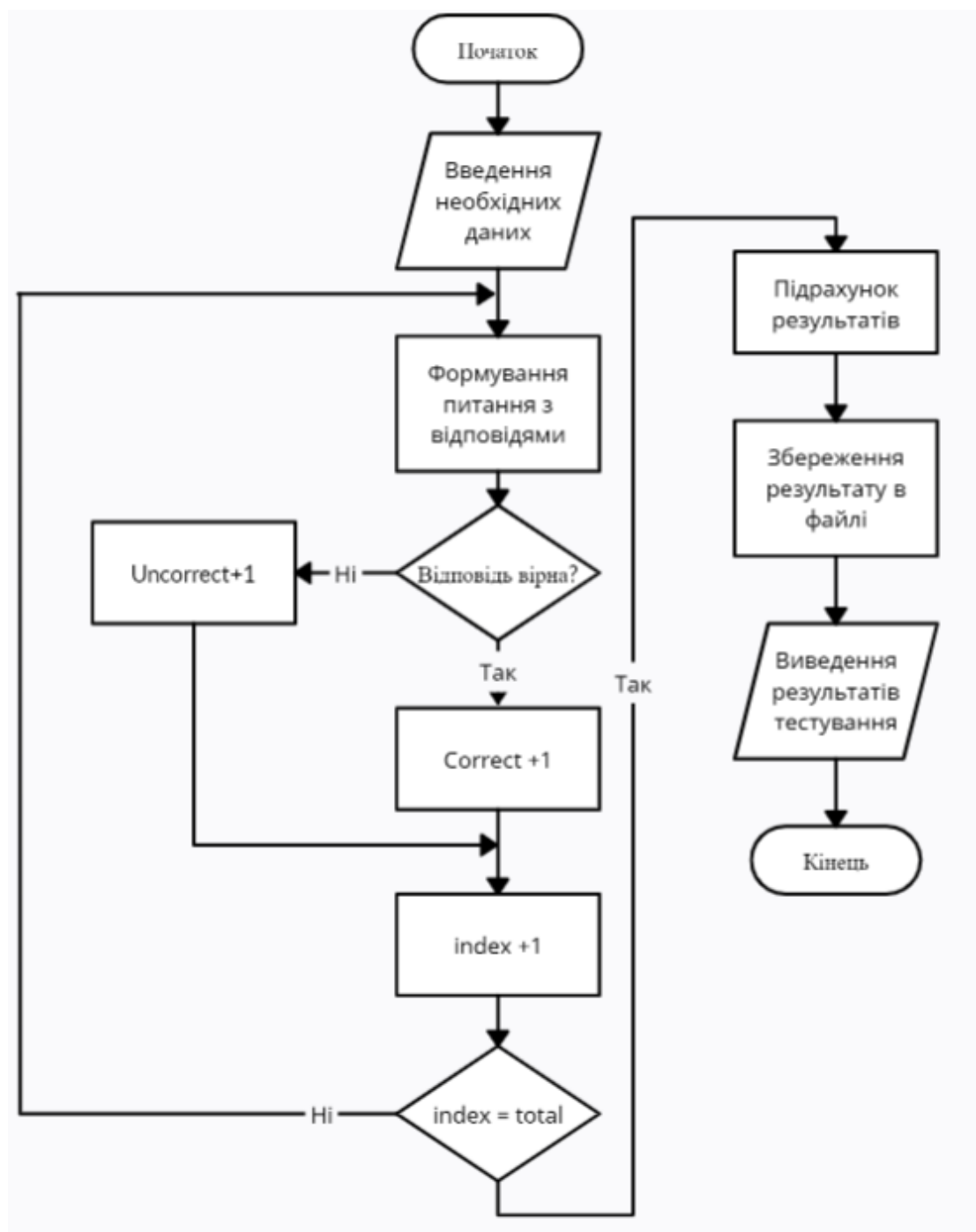


Рис. 2.6. Алгоритм проходження тесту

В якій використовуються такі геометричні елементи:

- еліпс – початок алгоритму і його кінець;
- паралелограм – являє блок вхідних або вихідних даних;
- прямокутник – блок обчислень даних і їх обробку;
- ромб – являє блок прийняття рішень.

Для проходження тесту необхідним буде ввести особисті дані, такі як: прізвище та ім'я, вибрати факультет і групу, а також білет по якому буде проходити тестування. Відповідно вибраному білету, сформується питання і відповіді до них, одна з яких вірна.

Далі користувач проходить тестування і при кожній відповіді на питання програма рахуватиме кількість правильних та неправильних відповідей. Коли питання в білеті закінчатся, програма підрахує результат, який буде записаний в зовнішній файл з даними про користувача, які він вводив перед тестуванням. І потім виведе ці результати на екран для користувача.

### 2.3.3. Розробка допоміжного функціоналу

При аналізі подібних додатків тестування, одним з недоліків було те, що результат тестування не зберігався автоматично, а потребував участі викладача. Тому в додатку необхідно розробити збереження результатів проходження тесту в зовнішній файл (Excel файл).

Існує велика кількість відкритих бібліотек, які дозволяють працювати з Excel файлами в Java. В нашому випадку, використовувався Apache POI, що являє собою популярний API, який дозволяє програмістам використовувати файли MS Office для запису і читання з них в Java додатках. Це бібліотека, яка має відкритий вихідний код, розроблена Apache Software Foundation.

Для її використання необхідно завантажити з офіційного сайту jar файли, і додати їх в середовище розробки (рис. 2.7) [13].

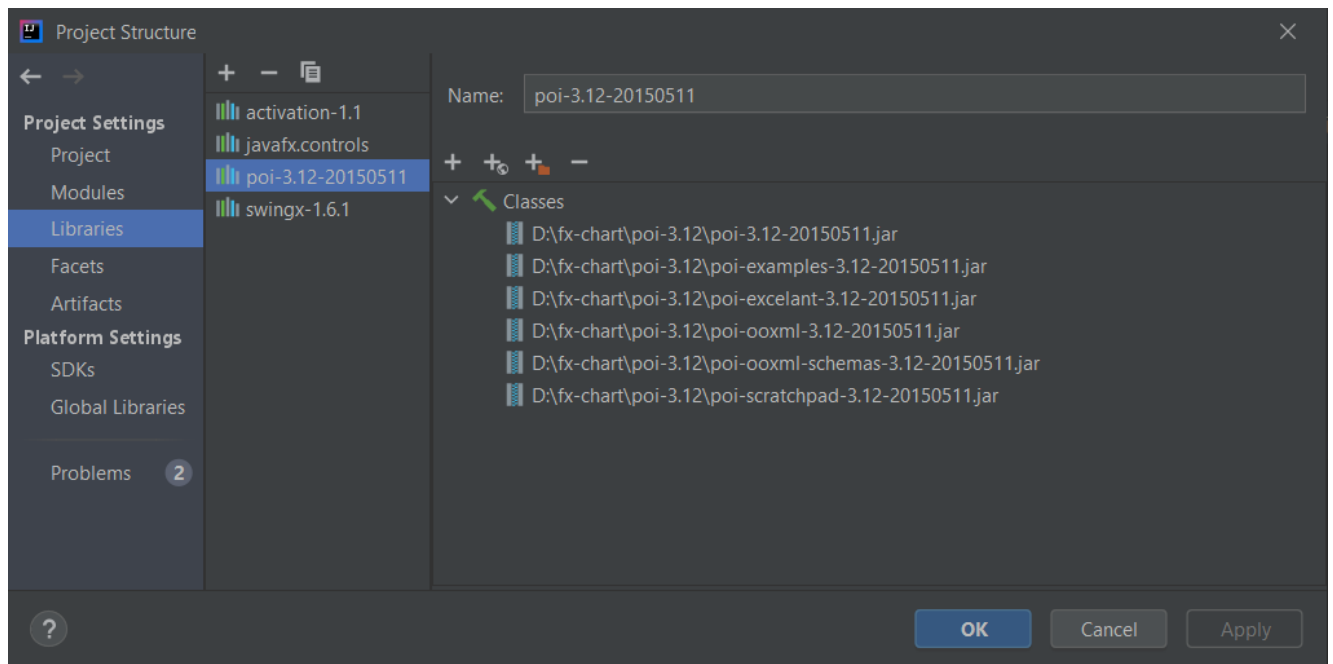


Рис. 2.7. Додавання бібліотек Apache POI

Для реалізації запису в Excel файл нам потрібно:

- створити об'єкт `HSSFWorkbook`;
- створюємо лист в книзі за допомогою метода `createSheet()`;
- за допомогою метода `createRow()` створюємо рядок;
- задаємо в рядку комірку – `createCell()`;
- записуємо workbook в File через `FileOutputStream`;
- закриваємо workbook за допомогою метода `close()`.

Запис даних в Excel.

```
Workbook workbook = new HSSFWorkbook();

Sheet newSheet = workbook.createSheet("Лист");
Row row = newSheet.createRow(0);
row.createCell(0).setCellValue("Імя");
row.createCell(1).setCellValue("Прізвище");
row.createCell(2).setCellValue("Факультет");
row.createCell(3).setCellValue("Група");
row.createCell(4).setCellValue("Квиток");
row.createCell(5).setCellValue("Таймер");
row.createCell(6).setCellValue("Результат");
row.createCell(7).setCellValue("Дата");
```



```

try {
    FileOutputStream fileOutputStream = new FileOutputStream("D:\\Mark.xls");
    workbook.write(fileOutputStream);
    fileOutputStream.close();
    try {
        FileInputStream myxls = new FileInputStream("D:\\Mark.xls");
        HSSFWorkbook studentsSheet = new HSSFWorkbook(myxls);
        HSSFSheet worksheet = studentsSheet.getSheetAt(0);
        int lastRow = worksheet.getLastRowNum();
        System.out.println(lastRow);
        Row row1 = worksheet.createRow(++lastRow);
        row1.createCell(0).setCellValue(sfname);
        row1.createCell(1).setCellValue(slname);
        row1.createCell(2).setCellValue(sfaculty);
        row1.createCell(3).setCellValue(sgroup);
        row1.createCell(4).setCellValue(stest_paper);
        row1.createCell(5).setCellValue(timer);
        row1.createCell(6).setCellValue(result);
        row1.createCell(7).setCellValue(date);
        myxls.close();
        FileOutputStream output_file = new FileOutputStream(new
File("D:\\Mark.xls"));
        studentsSheet.write(output_file);
        output_file.close();
    }
}

```

Важливим є також те, щоб користувач міг підтримувати зв'язок з адміністратором, щоб вирішити певні питання або вказати на помилки, які виникли під час роботи з додатком, щоб потім їх усунути.

Для цього в додатку необхідно реалізувати форму зворотного зв'язку, де користувач зможе написати повідомлення адміністратору на пошту.

Реалізація відправлення повідомлення до адміністратора.

```

public class EmailSender{
    public static void mailto(List<String> recipients, String subject,
        String body) throws IOException, URISyntaxException {
        String uriStr = String.format("mailto:%s?subject=%s&body=%s",
            join(",", recipients),
            urlEncode(subject),
            urlEncode(body));
        Desktop.getDesktop().browse(new URI(uriStr));
    }
}

```

```

}
private static final String urlEncode(String str) {
    try {
        return URLEncoder.encode(str, "UTF-8").replace("+", "%20");
    } catch (UnsupportedEncodingException e) {
        throw new RuntimeException(e);
    }
}
public static final String join(String sep, Iterable<?> objs) {
    StringBuilder sb = new StringBuilder();
    for(Object obj : objs) {
        if (sb.length() > 0) sb.append(sep);
        sb.append(obj);
    }
    return sb.toString();
}
public EmailSender(String selectedItem, String text){
    try {
        mailto(Arrays.asList("artem.varenick@gmail.com"), selectedItem,
            text);
    } catch (IOException e) {
        e.printStackTrace();
    } catch (URISyntaxException e) {
        e.printStackTrace();
    }
}
}
}

```

## Висновки до розділу 2

В другому розділі представлена розробка додатку тестування знань мовою Java за допомогою середовища розробки IntelliJ IDEA і описано основні етапи розробки.

Було розглянуто процес налаштування і підготовки середовища розробки і основних його елементів, що використовуються при написанні коду.

Описали функціональність системи за допомогою діаграми прецедентів, де виділили основних акторів, які працюють з системою і варіанти використання.

Також провели огляд найпопулярніших GUI фреймворків Java і вбудованого GUI Designer, що використовуються для реалізації графічного інтерфейсу

користувача. В розробці використовувався Swing, адже він має велику кількість графічних компонентів, що дозволяє розробляти багатофункціональний і красивий інтерфейс. І всі Swing-компоненти є платформонезалежними, що покращує кросплатформність додатку. Виділили основні вимоги яким повинен відповідати наш інтерфейс.

Розробили алгоритм проходження тестування, який представлений у вигляді блок-схеми, що спростило програмну реалізацію модуля тестування. Описали програмну реалізацію додаткових функцій в додатку.

## РОЗДІЛ 3

### ВІДОБРАЖЕННЯ РОЗРОБЛЕНОГО ДОДАТКУ ТЕСТУВАННЯ ЗНАНЬ

#### 3.1. Інструкція з використання програмного продукту

При запуску додатку перед користувачем з'являється початкове вікно (рис. 3.1), де йому потрібно обов'язково перед тестування ввести данні:

- ім'я та прізвище;
- вибрати з списку факультет і ввести номер групи;
- вибрати з списку білет по якому буде проходити тестування.

Рис. 3.1. Початкове вікно додатку

Кафедра КІТ (47)				НАУ 21 28 76 000 ПЗ			
Виконав	Вареник А.М.			Відображення розробленого додатку тестування знань	Літера	Аркуш	Аркушів
Керівник	Колісник О.В.					44	6
Консульт.					412 122		
Н-котрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						

Також є можливість вибрати таймер, який обмежує час відповіді на питання, а якщо відповідь не була дана за час, то питання рахується невірним.

Після введення всіх необхідних даних, користувач може розпочати тестування натиснувши кнопку “Розпочати”. І перед ним з’являється вікно в якому він може проходити тестування (рис. 3.2).

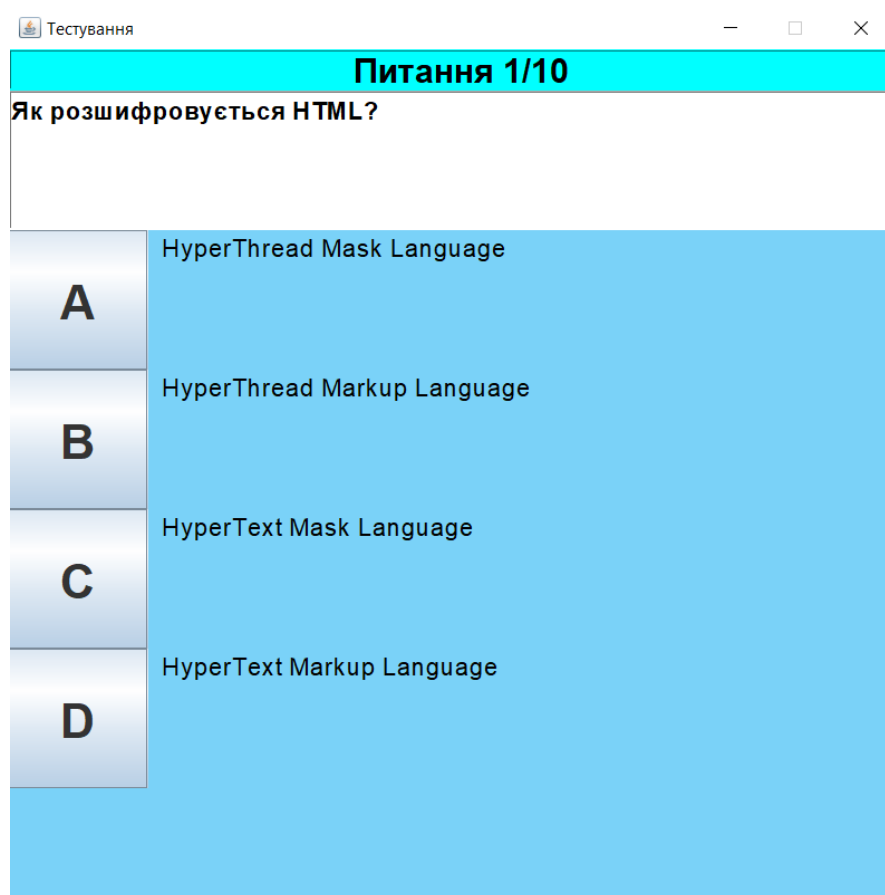


Рис. 3.2. Вікно проходження тестування

Вікно наповнюється контентом відповідно вибраному білету і має інформацію про номер питання із загальної кількості в білеті, трактування питання і відповіді до нього.

Після того як користувач закінчив тестування йому з’являється вікно (рис. 3.3) на якому він може побачити свій результат:

- загальна кількість питань на які було дано відповідь, кількість правильних і неправильних відповідей;
- оцінка за тестування.

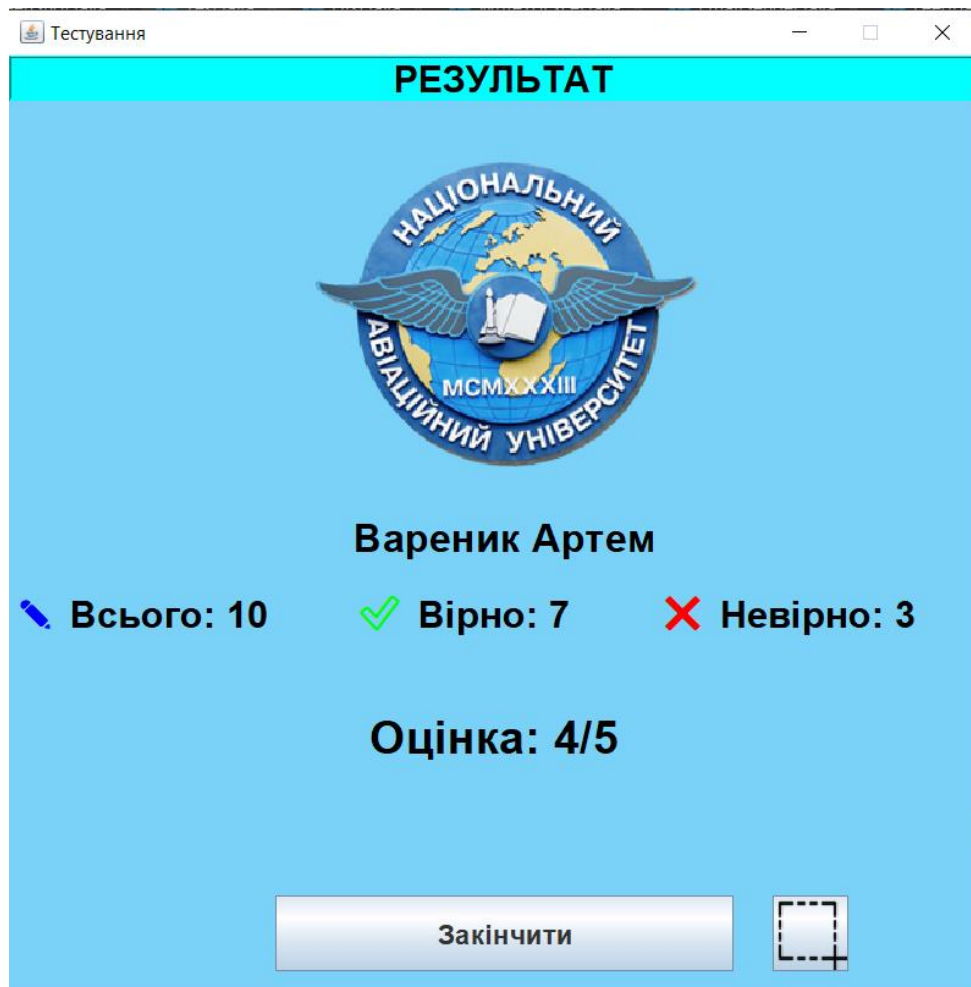


Рис. 3.3. Вікно результату проходження тестування

Після закінчення тестування дані про користувача і результат тестування зберігається в Excel файлі (рис. 3.4), який автоматично створюється при першому використанні додатку, а потім просто доповнюється новими даними. Також за необхідністю користувач може зробити знімок вікна з результатом просто натиснувши кнопку.

В Excel зберігаються такі дані:

- ім'я і прізвище;
- факультет і група;
- білет по якому проходило тестування;
- тестування проходило з таймером чи без;
- результат – фактична оцінка за тест;
- дата і час проходження тестування.

	A	B	C	D	E	F	G	H	I	J
1	Імя	Прізвище	Факультет	Група	Квиток	Таймер	Результат	Дата		
2	Артем	Вареник	ФККПІ	412	Білет №1	off	5/5	2021-05-17T14:15:22.935231201		
3	Артем	Вареник	ФККПІ	412	Білет №1	off	5/5	2021-05-17T14:21:31.952895900		
4	Юлія	Остоматій	ФККПІ	412	Білет №1	on	5/5	2021-05-17T14:25:57.114940600		
5	Артем	Вареник	ФККПІ	412	Білет №2	off	4/5	2021-05-17T14:30:50.135343600		
6										
7										

Рис. 3.4. Збережені данні тестування в Excel файлі

Також якщо у користувача при роботі з додатком виникнуть якісь зауваження або питання до розробника/адміністратора він може сформулювати йому повідомлення напяму в додатку (рис. 3.5).

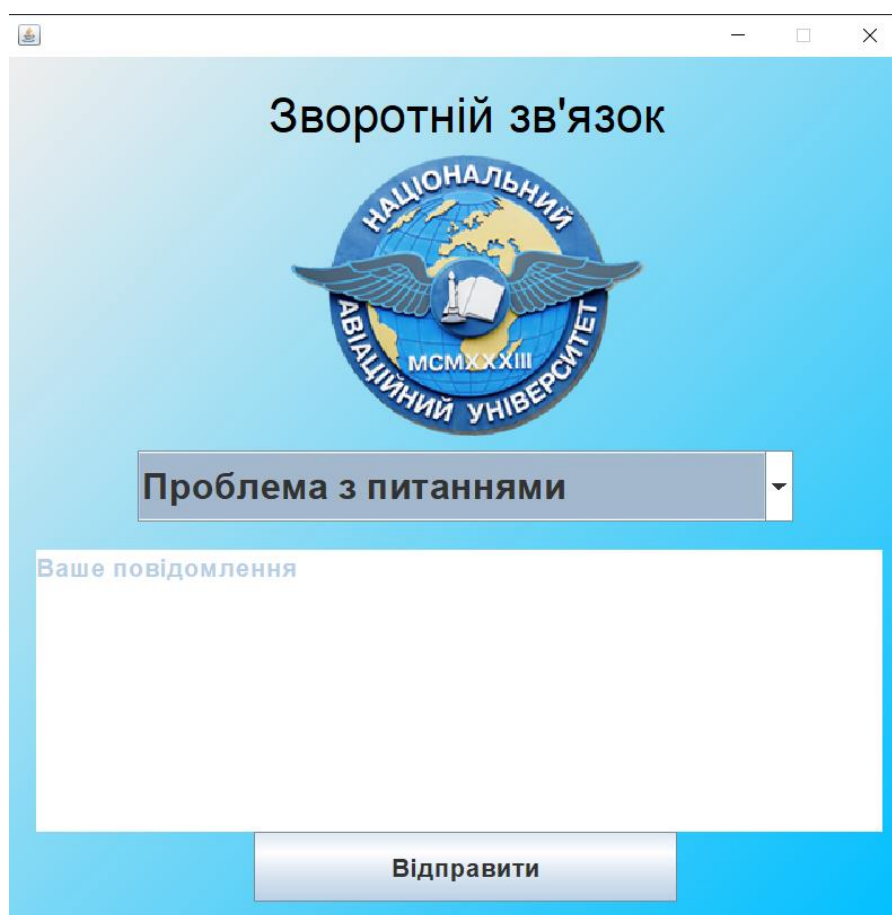


Рис. 3.5. Вікно зворотного зв'язку

Інтерфейс додатку простий, логічний і зрозумілий, а також захищений від помилок користувача під час роботи з ним, адже реалізовані спливаючі повідомлення (рис. 3.6), що інформують про некоректність дій.

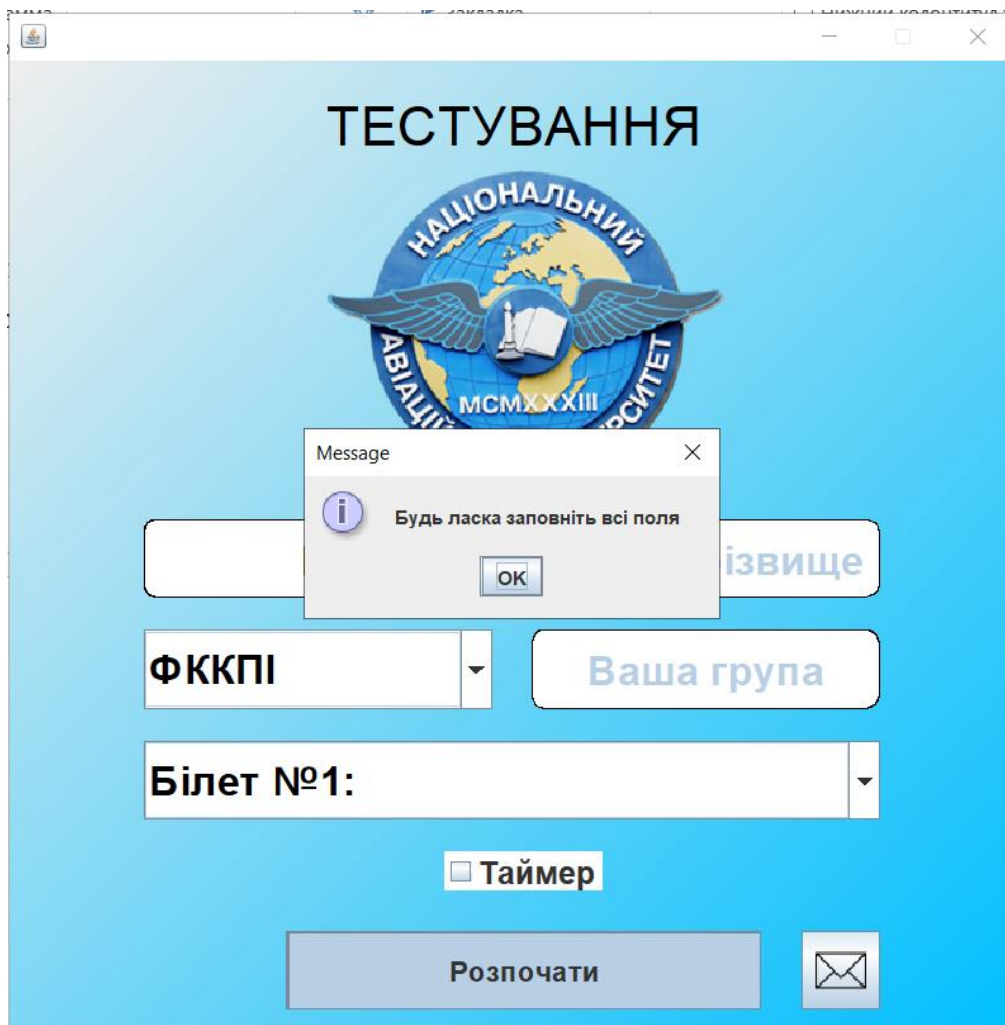


Рис. 3.6. Спливаюче повідомлення

Також при наведенні на елементи інтерфейсу існують підказки, що робить інтерфейс більш зрозумілим, адже користувач розуміє, що йому потрібно зробити при взаємодії з тим чи іншим графічним елементом. Наприклад, якщо навести на поле введення імені, користувачу висвітлиться підказка (рис. 3.7).

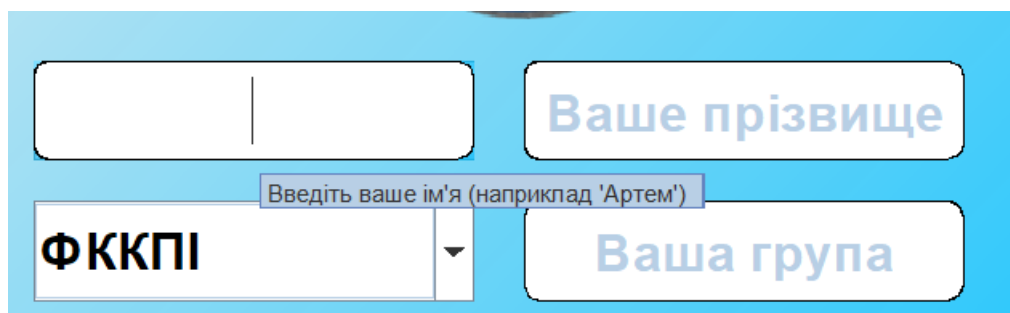


Рис. 3.7. Підказки



### **Висновки до розділу 3**

Третій розділ відображає опис інтерфейсу розробленого додатку і інструкція з його використання. Розроблений інтерфейс відповідає всім поставленим вимогам, а за допомогою реалізованих спливаючих повідомлень і підказок мінімізує виникнення помилок у роботі з додатком.

Додаток зручний і простий у користуванні і має весь необхідний функціонал для проведення процесу тестування і підвищення його ефективності.

## ВИСНОВКИ

Підсумовуючи все вищевикладене, можна зазначити, що для досягнення мети у дипломному проекті були виконано у повному обсязі усі поставлені задачі:

1. Проаналізувати існуючі додатки тестування. Був зроблений огляд одних з найпопулярніших програм тестування: MyTestX, Айрен, EasyQuizzу і визначили переваги і недоліки кожної. Основним недоліком оглянутих систем є те, що вони сумісні тільки з ОС Windows і дещо функціонально перевантажені.

2. Огляд мов програмування і середовищ розробки та здійснити їх порівняльний аналіз. В ході виконання дипломного проекту, оглянули найпопулярніші мови програмування загального призначення: Java, C#, Python, C++. Для проведення порівняльного аналізу виділили базові вимоги. Кожна мова має свої переваги і недоліки, але для реалізації було вибрано Java, оскільки вона забезпечує легку роботу з GUI і 2D графікою, програми дешево розробляти та обслуговувати, а також додатки написані на цій мові не залежать від платформи.

Відповідно вибраній мові програмування, розглянули найпопулярніші інтегровані середовища розробки: Eclipse і IntelliJ IDEA. Перевагу було віддано останньому, тому що він зручний і простий у використанні, має вбудований GUI Designer і його функціонал повністю задовольняє потреби для розробки.

3. Огляд фреймворків для реалізації інтерфейсу користувача. Було оглянуті стандартні фреймворки для реалізації інтерфейсу користувача в Java: Swing і AWT. Для розробки був обраний перший, оскільки простий у використанні і має велику кількість графічних компонентів, за допомогою яких можна створювати красивий інтерфейс з широким функціоналом.

4. Реалізувати інтерфейс користувача. Розробка інтерфейсу користувача проводилась за допомогою GUI фреймворка Swing і вбудованого в середовище розробки GUI Designer, який спростив і пришвидшив розробку.

5. Розробити модуль тестування. Було розроблено алгоритм проходження тестування, що потім спростило програмну реалізацію модуля за допомогою можливостей мови програмування Java.

Розроблений додаток слугує для розширення можливостей контролю і оцінювання рівня знань, а також для підвищення якості освітніх процесів. Він може використовуватися для:

- актуалізації певних знань;
- контролю поточних, проміжних і підсумкових знань;
- проведення самоконтролю.

Додаток може забезпечити просту і швидку процедуру проведення тестування, що зекономить час і дасть можливість протестувати велику кількість студентів одночасно.

Також за допомогою додатку стає можливим знизити витрати на проведення тестування, адже не потрібно роздруковувати бланк тесту, а достатньо сформулювати “банк тестів”, які потім просто додати в додаток.

Але слід зазначити, що комп’ютерне тестування необхідно поєднувати з іншими методами перевірки і форм контролю рівня знань, для забезпечення максимально точного оцінювання.

## СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Компьютерное тестирование знаний MyTestXPro [Електронний ресурс] – Режим доступу: [http://mytest.klyaksa.net/wiki/Заглавная\\_страница](http://mytest.klyaksa.net/wiki/Заглавная_страница) (дата звернення 15.05.2021р.). – Назва з екрану
2. Программа тестирования знаний Айрен [Електронний ресурс] // Офіційний сайт Айрен – Режим доступу: <https://irenproject.ru> (дата звернення 17.05.2021р.). – Назва з екрану
3. Конструктор тестів easyQuizzу [Електронний ресурс] // Офіційний сайт Easyquizzу – Режим доступу: <http://easyquizzу.ru> (дата звернення 18.05.2021р.). – Назва з екрану
4. Нимейер П. Программирование на Java / П. Нимейер, Д. Леук; [пер. с англ. М. А. Райтмана]. – Москва : Эксмо, 2014. – 1216 с.
5. Джекпкс Ф. Язык программирования C# 7 и платформы .NET и .NET Core / Ф. Джекпкс, Е. Троелсен. – Москва : Диалектика, 2020. – Т. 1. – 672 с.
6. Лутц М. Изучаем Python / М. Лутц. – Москва : Диалектика, 2020. – Т.1. – 832 с.
7. Страуструп Б. Программирование: принципы и практика использования C++ / Б. Страуструп. – Москва : Вильямс, 2011. – 1248 с.
8. Інтегроване середовище розробки Eclipse [Електронний ресурс] // Офіційний сайт Eclipse Foundation – Режим доступу: <https://www.eclipse.org> (дата звернення 22.05.2021р.). – Назва з екрану
9. Інтегроване середовище розробки IntelliJ IDEA [Електронний ресурс] // Офіційний сайт JetBrains – Режим доступу: <https://www.jetbrains.com/ru-ru/idea/> (дата звернення 23.05.2021р.). – Назва з екрану
10. Java Development Kit [Електронний ресурс] // Сайт Oracle – Режим доступу: <https://www.oracle.com/java/technologies/javase-jdk16-downloads.html> (дата звернення 23.05.2021р.). – Назва з екрану

11. Java AWT Tutorial [Електронний ресурс] – Режим доступу: <https://www.javatpoint.com/java-awt> (дата звернення 25.05.2021р.). – Назва з екрану
12. Java Swing Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javatpoint.com/java-swing> (дата звернення 26.05.2021р.). – Назва з екрану
13. Apache POI [Електронний ресурс] – Режим доступу: <https://poi.apache.org> (дата звернення 04.06.2021р.). – Назва з екрану